DOCUMENT RESUME

ED 304 083 IR 013 601

TITLE	Public Domain Software for Education.
INSTITUTION	Scholastech, Inc., Cambridge, MA.
SPONS AGENCY	Fund for the Improvement of Postsecondary Education (ED), Washington, D.C.
PUB DATE	86
GRANT	G008541272
NOTE	23p.
PUB TYPE	Reports - Research/Technical (143)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Artificial Intelligence; *Courseware; Curriculum Development; Database Management Systems; Educational Improvement; Electronic Mail; *Evaluation Criteria; Higher Education; *Information Dissemination; *Information Networks; *Library Collection Development; Microcomputers; Needs Assessment; User Needs (Information)
IDENTIFIERS	*Free Materials; *Public Domain Software; Software Evaluation

ABSTRACT

This report describes a project undertaken by Scholastech, a public charity addressing the development needs of computing educators, which was designed to advocate and support the increased use of free software in the college curriculum. The first of five sections provides a brief overview and statement of project objectives, i.e., to serve college instructors, small college administrators, and educational researchers searching for cost-effective, educationally-tested software for the curriculum. The second section focuses on the purpose of the project, which was to define the corpus of software in the public domain, determine the role of this free software in the curriculum, identify reliable sources of free educational software, and consider the problem retrospectively. The background and origins of the project are discussed in the third section, and the fourth presents descriptions of the major project activities, including the development of a collection of free software (MS-DOS, UNIX, Macintosh), the formalization of software evaluation criteria, the creation of a database for the management of software evaluations, the development of a delivery system in the form of the mass distribution FIPSE (Fund for the Improvement of Postsecondary Education) disk, and the distribution of the FIPSE R&D Disk. The final section reports the results of the study, which suggest that schools can afford to provide commercial hardware and software, or pursue major development projects, but not both. Suggested evaluation criteria, a sample "Quick Reference Listing," and sample resource screens for IBM/ATT PC-compatibles are appended. (CGD)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED304083

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

COVER SHEET

Grantee Organization:

Scholastech Inc.
P.O. Box 1545
Cambridge MA 02238

Grant Number:

G008541272

Project Dates:

Start: December 1985
Ending: December 1986
Number of months: 12

Project Director:

Dr. Stanley Doherty
Executive Director
Scholastech Inc.
P.O. Box 1545
Cambridge MA 02238

Fund Program Officer:

John Donahue

Grant Award:

$ 17,911

IR013601

BEST COPY AVAILABLE

Project Summary

Public Domain Software For Education

Scholastech Inc.

Scholastech is a public charity addressing the development needs of computing educators. To advocate and support the increased use of free software in the college curriculum, Scholastech *collected* and *evaluated* some 500+ free software packages relevant to education (MS–DOS, UNIX, Macintosh), *distributed* 1600+ free diskettes profiling the results of its research, and has *supported* a free electronic bulletin board archiving all its resources.

Dr. Stanley Doherty
Project Director
Scholastech Inc.
P.O. Box 1545
Cambridge MA 02238

IR013601

1.0 Project Overview

Scholastech Inc. is a public charity serving the development needs of computing educators. As an organization we are committed to the ideal that curriculum computing should be as cost–effective as it is learning–effectual. The Scholastech "Public Domain Resources For Education" Project developed as an outgrowth of our organization's standing services for computing educators. Prior to the formulation of our FIPSE grant, Scholastech collected public domain resources, evaluated them for their educational merit, and distributed them at cost to schools and universities. With FIPSE funding we wished to make these standing activities more comprehensive, formal, and effectual. For every MS–DOS or Macintosh program which we acquired and tested, we knew that there were four more available on bulletin boards, in user groups, or at universities. To present a coherent argument for the use of free software in the curriculum, Scholastech needed to speak with the authority of experience. Toward that end we applied for our FIPSE grant.

Our project was designed to serve three populations searching for cost–effective, educationally–tested software for the curriculum:

a. ***College instructors:*** teachers who are looking to integrate the computer with existing courses but, for lack of affordable software, stay on the sidelines.

b. ***Small college administrators:*** decision–makers and administrators at community colleges and adult education programs hesitant to commit scarce resources for hardware if there were not reliable sources for affordable software.

c. ***Educational researchers:*** scholars and administrators in the field of educational computing who were wary of the quality and legal status of public domain and shareware software.

In course of our project we acquired some 541 public domain and shareware programs for MS–DOS PCs, Apple Macintoshes, and UNIX systems. We were able to evaluate all of the significant programs and most of the smaller utilities in light of our experience as full–time professors and researchers. To our knowledge, no comprehensive evaluation of public domain resources for education had been conducted thus far or since. To disseminate the results of our research, we decided to develop and distribute an MS–DOS diskette containing the following:

1. A description of our project goals, methodology, and results

2. A listing of national user groups distributing public domain software

3. A critical listing of 200 free bulletin boards offering significant collections of public domain software

4. A listing of the programs which Scholastech found most useful and functional for curriculum computing

5. A small collection of public domain programs
   - ** EMACS: a multi-windowing text editor
   - ** KERMIT: a communications program
   - ** 3BY5: a data base manager
   - ** UTILITIES: for disk management, printing, and text processing

Beyond Scholastech members involved in collecting, evaluating, and redocumenting these resources, we received the support of a number of local institutions and professional organizations. Our early versions of the distribution disk were tested and critiqued by professors, administrators, and students in the Boston area. Over 1600 free copies of the resource disk were subsequently distributed at three national conferences (NECC '87, EduCOM, IBSCUG) and at six local conferences.

Many of the educators receiving this disk subsequently contacted Scholastech regarding further support, updates, or new submissions. These educators said that they benefitted from the disk in the following ways:

a. *COMMUNICATIONS*: They used our listing of bulletin board numbers and the KERMIT program to access public networks and to obtain free software.

b. *COLLECTIONS DEVELOPMENT*: They used our evaluation criteria and list of recommended software as the groundwork for on-campus collections of public domain resources.

c. *UTILTIES*: They copied some of the disk utilities for students and colleagues.

Without a systematic mechanism for evaluating the impact of our project, we have to rely upon the responses of educators contacting us by mail or via our computer bulletin board. Most teachers with modems went right to the recommended bulletin boards and started downloading software for their students. Teachers without modems contacted local or national user groups. Administrators were able to build substantive libraries of public domain software so that any faculty member or student on campus could obtain free word processors, spreadsheets, data base managers, and communications programs. In essence, we perceived that many of the educators using our resource disk gradually assumed the role of public domain librarians at their respective schools.

2.0 The Purpose of the Scholastech Project

Short on funds and in need of software which we could copy within the organization, Scholastech discovered software in public domain. To our amazement, we discovered moreover that some of the free programs outperformed the packages costing our colleges $200-300. Considered as cost-effective resources for poor

colleges or as high–performance resources for development, free software seemed to be a fantastic resource without vocal advocates in the educational community. In our efforts to introduce other educators to the benefits of public domain and shareware educational software, we defined three related tasks: determining the existing corpus of free software for education, clarifying for educators how free software could be used in the curriculum, and defining the more reliable sources of public domain software.

2.1 Defining the Corpus of Free Software: "What's Out There?"

When Scholastech first began collecting MS–DOS free software in 1983, there were relatively few distributors and very few programs of professional quality. For educators to take free software seriously, it seemed imperative for us to locate serious software, packages which would refute the addage "You get what you pay for." Although we managed to collect and test dozens of packages in the next two years, we were unable to develop our library systematically. For every new program acquired and tested, there were four or five new ones appearing on bulletin boards and in user groups. To complicate things, many of the better programs were distributed as shareware, i.e. software distributed freely upon the condition that the user send the developer money if the program proved to be useful. To bolster confidence in shareware software, we needed to survey the conditions which developers placed on their packages. Which shareware conditions permitted professors to copy software for a class? Which shareware conditions prohibited such distribution?

Without a comprehensive vision of "what was out there," educators would remain tentative in acquiring free software. They would opt for the apparent security of a $125 word processor over a free one of equal capability.

2.2 Determining the Role of Free Software in the Curriculum

As a related problem, we needed to demonstrate that free software could assume a number of roles in the curriculum. It had been our experience that the availability of quality free software encouraged dialogue between faculty and students, fostered in–house resource development, and clarified any cost–benefit analysis. Our experiences with free software would not necessarily be relevant to the average college or adult learning program. We needed to distribute to educators a sampling of free software and to assess how this sampling was used. Did the availability of free word processors encourage faculty to make copies for students in writing courses? Did the availability of math programs and equation solvers encourage math professors to develop in–class applications? Would the formation of public domain libraries encourage faculty to experiment more freely with multiple resources? Could such a library of public domain software support many, if not most, applications in the freshman–sophomore curriculum?

If the experiences of educators using our public domain collections confirmed our own experiences, then we could argue to the educational community at large that free software should have some role in the curriculum or at the very least should

be considered prior to the selection of commercial software. This argument had to be grounded upon the experience of many educators teaching in a variety of educational cultures.

2.3 Determining Reliable Sources of Free Educational Software

When we framed our preliminary proposal in November 1984, there were few norms in the field of collegiate microcomputing. Some colleges were running AppleIIs, others CP/M machines, and yet others MS–DOS. Educators using a Wang PC under MS–DOS could not run software configured for the DEC Rainbow or IBM PC. In this fragmented environment, educators using one type of PC needed to link up with collections of free software configured specifically for their computer. Further, educators needed to link up with reputable distributors of free software. Recall that the number of bulletin boards and clandestine user groups distributing pirated copies of commercial software was at an all–time high. A professor distributing commercial software would be inviting a lawsuit. Even if a greater number of educators and administrators were willing to work with free software, their ambitions would be thwarted by the lack of reliable sources. Educators needed to be confident that a program would run on their PC and that it was genuinely free software.

2.4 The Problem Considered Retrospectively

Between the time of our preliminary proposal (November 1984) and eventual funding (December 1985) the complexion of curriculum computing changed dramatically. Market pressures toward standardization in the office gradually encouraged standardization in the curriculum. Most schools adopted IBM PC–compatibles or the new Apple Macintosh. With this hardware standardization came a certain degree of software standardization. Without evaluating free resources, many colleges automatically adopted such commercial packages as *Wordstar*, *WordPerfect*, *Lotus 123*, *Multiplan*, and *Turbo Pascal*. Colleges invested considerable monetary and human resources in training academics how to use these "standard" packages. In practice, curriculum computing was more frequently approached as an issue of resource management and not one of learning or resource development.

Scholastech had framed much of its argument for free software upon common sense––"students should not pay for software if equally–capable packages were available for free." This was naive politically. Common sense also dictates that if a department saves money one year by selecting public domain software, its budget for the following year would be cut or reallocated accordingly. To advocate free software in budget–conscious departments, Scholastech has had to develop an entirely separate argument based upon instructional development––"the open architecture of free software offers educators greater power and flexibility in designing discipline–specific applications." In the course of our project, we began to evaluate all our programs for their performance and for their development potential. This shift in emphasis has, in some respects, more far–reaching implications for curriculum computing than the argument for cost–effectiveness. If we can demonstrate to educators that closed–architecture *WordPerfect* supports far less

instructional development than such open–architecture programs as *PC–Write*, *New York Word*, or *MicroEMACS*, we may encourage educators to experiment with the development of applications.

3.0 Background and Origins

Although educational institutions occasionally dabbled in free software programs, Scholastech saw little evidence that colleges were willing to approach free software with the same professional acumen that they brought to the evaluation and selection of commercial packages. Prior to our FIPSE funding Scholastech had developed a reliable and professional process of acquiring and evaluating free software. As the primary goal for our project, we wished to demonstrate to educators nationwide that our approach could serve as an initial model for their own in–house evaluations. As an organization composed of full–time professors and computer professionals, Scholastech was not equipped to become a national clearinghouse for free educational software. Rather we wished to assume a leadership role in setting guidelines for the evaluation and use of free software regardless of source or distributor. There were, after all, dozens of newsletters and educational user groups discussing the use of *Wordstar* and *Lotus 123*, but virtually no group addressing the educational benefits of public domain work–alikes (e.g. *MicroEMACS, ASEASYAS*). In short, we wished to offer the educational community a replicable model for developing in–house public–domain libraries analogous to the ones which Scholastech has developed and maintained over the years. If we could show professors and administrators exactly how to acquire and evaluate free software, we would have achieved our primary goal. The process of selecting appropriate free software for various curricula would have to be left to the educators in that curriculum––not to an outside group like Scholastech.

To insure that our own model could be replicated in the greatest number of institutions, we made the following assumptions about our audience:

1. The educators served by our project had little if any money to spend on software.
2. The educators had access to common personal computers like the IBM PC or Apple Macintosh.
3. The educators were not particularly computer–literate. They would not have to know anything about programming or hardware or operating systems.
4. Our educators would be willing to spend some time evaluating programs which they acquired.
5. Our educators were not primarily interested in tutorials or little flash–card programs. Our supporters to date were most interested in major packages: word processors, data base managers, languages, spreadsheets, graphics editors, expert systems, and

development utilities. The programs were educational by virtue of their support for learning and support for curriculum development.

6. Our educators would be willing to share their discoveries with Scholastech and other educational computing groups.

Further, we were determined to design our mass distribution disk so that a user with no computer expertise would be able to learn about free software and acquire free software. It was our experience that the educators most in need of educational software were often the ones least familiar with the technical intricacies of accessing bulletin boards and evaluating program performance. Our assumptions going into the FIPSE project were patently populist--free software for democratic society. We had demonstrated to the satisfaction of our local supporters that free software could play a major role in curriculum learning and curriculum development--we simply needed funding in order to concretize our experience and resources in a form easily understood and easily used by the average teacher.

4.0 Description of Major Activities

As mentioned earlier, Scholastech had collected a significant library of free software for education, but it was by no means an exhaustive collection. Our evaluation process, moreover, was largely informal. As teachers, we looked for a number of key features in software programs that would enhance student learning. We did not articulate what those features were in a global sense until we had to develop the FIPSE materials. Our primary activities under the grant were responses to these needs. To define the corpus of free educational software, we had to develop a more comprehensive collection. To provide educators nationwide with a reliable evaluation model, we needed to articulate and test our informal evaluation criteria. To make all of this information comprehensible to the average teacher, we needed to develop and test a delivery system--the mass distribution FIPSE disk.

4.1 Collection Development

We began the project with a standing collection of some 200 MS-DOS, 15 UNIX, and 21 Macintosh programs. Although many of these programs were excellent, we could not be certain that they were the best examples of free software for education. In the first two months of the project our group of six professors accessed some 87 bulletin boards and contacted 23 user groups in an effort to acquire any and all free software with the least potential for educational development. All in all we acquired another 430 MS-DOS programs, another 44 UNIX programs, and another 67 Macintosh programs. Regardless of operating system, we initially categorized each program by its apparent function:

a. artificial intelligence
b. communications
c. data base management

d. development utilities and languages
e. file management
f. graphics editors and utilities
g. quantitative analysis (math, statistics)
h. source code (anything in C, Pascal, Fortran, LISP, Prolog, etc.)
i. text processing (editors, word processors, utilities)

As you might expect, there was tremendous redundancy within any category: 44 editors, 26 spreadsheet programs, and 60+ sort utilities. Prior to applying our formal evaluation criteria, we discovered that some of the programs did not work or worked unreliably. These programs were removed from the evaluation library immediately.

For those programs distributed in original source code, we attempted to recompile the most promising of them on a variety of machines (e.g. *EMACS, Kermit, SDB Database*). This aspect of the project was most interesting for it illustrated that there could be a certain number of programs running on every type of college computer--the PC, Macintosh, Amiga, Vax, UNIX PC, et al.. To be certain, a basic screen editor was not as impressive on the Macintosh as *MacWrite*, but then again the Macintosh user would not have to learn five different editors when using a PC, Vax, or UNIX system on campus. A greater degree of software coherence could be achieved with free software than with commercial software. The more complex and differentiated the college network, the more such coherence is needed.

4.2 Software Evaluation Criteria

We wished to impress our audience with the differences between descriptive and prescriptive evaluation criteria. It was not our intent to recommend a select number of programs (the 'greatest hits'), but to teach educators how to conduct their own evaluations, how to determine whether a program was complex or simple, friendly or difficult. Toward that end we formalized our evaluation criteria into five areas:

a. Hardware Issues
- ** Compatibility: how many PCs and operating systems would run the program?
- ** Installation: was the program easy to set up?
- ** Peripherals: how well did the program support various printers, modems, graphics pads, etc.
- ** reliability: were there causes for program failure?

b. Categorical Issues
- ** Basic features common to editors, spreadsheets, data base managers, etc..
- ** Advanced features common to editors, spreadsheets, etc..

c. User Issues
- ** Documentation: how much? how detailed? how understandable?
- ** Help screens: how many? how useful?
- ** User interface: commands, keys, icons, error-handling, menus, etc..
- ** Quirks: odd features, missing features

d. Development Issues

** Source code: included? available? well-structured?
** User-defined macros? templates? stylesheets? help files?
** Instructional development?

c. Legal and Support Issues
** Legal status: public domain? freeware? shareware?
** Support: phone? mail? bulletin boards?
** Registration: suggested? required?

Although these criteria were not exhaustive, they were clear. Teachers using the criteria subsequently told us that the checklist served as a starting point from which they could develop more specific criteria relevant to their student body, computer systems, and disciplines. Teachers from different departments shared a common vocabulary in evaluating programs. Many educators requested and received a printout of our entire evaluation file.

4.3 Evaluation Management

Beyond evaluation criteria, we wished to provide educators with a mechanism for collecting and managing software evaluations. If a computer center develops a library of free software, there should be a way for faculty and students to record and compare their responses to these programs. In this manner a public domain software library can serve as a focal point for the development of campus standards.

To evaluate all the free programs collected in the initial phase of our project, we developed a UNIX data base program and interactive questionnaire. As our evaluation team tested each program, the evaluator would access the Scholastech bulletin board, complete an evaluation questionnaire, and make a summary judgement on the value of the program. In this manner we were able to monitor the course of the evaluation process from day to day. Programs evoking mixed responses were tested by third and fourth evaluators. At any point we could search this database of evaluations for recurring patterns and discrepencies. This data base program, interactive questionnaire, and communications software has been released to the public domain and distributed to schools running compatible UNIX hardware. It is our hope to expand this on-line service so that evaluations of our entire collection would be accessible to educators from our bulletin board on a permanent basis.

4.4 Development of the Distribution Disk

Published reports are nice, but nothing encourages people to explore free software quite like giving them a free disk. From the start of our project we intended to develop an MS-DOS disk which would --

a. introduce educators to free software
b. introduce our evaluation criteria for free software
c. introduce educators to sources of free software (user groups, bulletin boards)
d. provide a sample catalogue of free software successfully used at colleges
e. report the results of our research

f. include a sampling of free software

The third major phase of our project concerned the development and selection of materials for this mass distribution disk. By squeezing the programs and support files into archives, we were able to fit 2.3 disks worth of material onto a single diskette. Unsqueezing the programs posed a few problems for our test group, but the process succeeded admirably in the long run.

I have attached a number of printouts (i.e. screen dumps) from the FIPSE R&D DISK, but the best way to evaluate its contents is to run it (**A>welcome {CR}**).

4.5 Distribution of the FIPSE R&D DISK

We had planned originally to mail all these disks to educators on our various mailing lists. As we clarified our purpose as advocates of free software, it occurred to us that more personal contact and distribution would better serve the educators receiving the disk. Consequently Scholastech members set up distribution booths at both national and local educational computing conference. At the NECC '87 conference at San Diego, for example, we talked with over 400 educators and distributed over 600 disks. The personal contact made all the difference for those teachers unfamiliar with the benefits of free software.

The FIPSE R&D DISK attached to this report is one of over 1600 distributed at conferences, at user groups, and in the mail. We encouraged all educators to make copies of the disk and to distribute it to friends. Thousands of copies are currently in circulation; Scholastech regularly receives mail from educators who have received a copy of a copy of a copy.

4.6 Project Participants and Evaluation

We designed our "Public Domain Resources For Education" project to be tightly-focused. To the extent that most of the project activities represented extensions of our ongoing services, the number of participants directly involved in development and evaluation was small: Dr. Stan Doherty (project director), Dr. James Adair (testing), Dr. William Buchholz (documentation), William Burton (programming, support), Dr. Jan Harrington (collection management), Thomas Kilburn (programming, evaluation), Scott Samenfeld (distribution). As we developed our materials and evaluations for distribution, however, we invited twenty colleagues from local colleges to evaluate test versions of the FIPSE R&D disk. Their responses were most encouraging; the disk provided them with a tremendous amount of useful information on free software. Our initial reviewers also discovered problems. For educators wholly unfamiliar with the operation of an MS-DOS computer, our test disk offered few instructions on how to view files or unsqueeze archives.

In response to these observations we developed an easy-to-use introductory file which explained the contents of the disk and necessary commands for viewing and

and unsqueezing files. Without these initial reviews we might have distributed a disk beyond the expertise of novice computing educators.

Given a modest grant, $17,900, it was not possible to develop a formal mechanism for evaluating the impact of our project. The educators receiving our FIPSE R&D DISK were all over the country in as many as 1100 institutions. We encouraged educators using the disk to get back in touch with us and to form an informal network for sharing software and evaluations.

5.0 Project Results

The "Public Domain Resources For Education" project produced two sets of deliverables: an authoritative argument and a resource network.

5.1 The Argument For Free Software

Although some schools have made remarkable progress in the field of computer-based learning, most researchers would conclude that the vision of a computer-literate, computer-intensive curriculum has not been realised at most schools investing in computers. Although the reasons for this disappointment are legion, we can be certain that the hidden costs of providing hardware and software across the curriculum inhibited the development of learning-intensive programs. If our research in the field proves accurate, it would seem that schools could afford to provide commercial hardware and software OR they could pursue major development projects but they could rarely manage both. In this context, the need for cost-effective software seems all the more urgent.

Based upon our FIPSE-funded research, Scholastech can advance and substantiate the following arguments:

a. There is a sufficient corpus of free MS-DOS, Macintosh, and UNIX software in the public domain to support most, if not all, software needs in the freshman-sophomore curriculum. Upper-division computing may often benefit from free software, but there are relatively few programs addressing advanced problem-solving.

b. Free software can be equal or superior in quality to commercial software packages in the areas of communications, data base management, quantitative analysis, and text processing. In the areas of graphics, programming, and artificial intelligence, a school may have to supplement free software with commercial packages.

c. Colleges and extension programs can establish and maintain a library of free software with a minimal initial investment (e.g. $200 for 60+ excellent MS-DOS programs).

d. Free software encourages educators to develop and distribute original instructional applications. Professors are more willing to develop data bases or text tutorials if they are able to distribute the program along with the instructional material.

e. A free software library on campus encourages interdisciplinary evaluation of computing resources and interdisciplinary development.

f. The availability of a large corpus of free software clarifies which development projects warrant funding for original programming.

Many departments request funding or support for development projects which duplicate programs and/or instructional environments already available in the public domain. If the corpus of available free software were made known to more colleges, less money would be wasted on redundant development.

5.2 The Resource Network

It has been our experience that once educators are exposed to quality freeware, they are more than willing to share their discoveries and frustrations freely. Unlike commercial environments supporting a fee–for–service mode of interaction, freeware encourages a fee–for–satisfaction dynamic. If a shareware program does not address an educator's need, he or she has lost nothing. If the public domain or shareware program assists the educator, he or she is usually motivated to discuss why it helped. As Scholastech distributes or reviews more freeware from year to year, we develop a very active and responsive constituency.

Appendix A: Conference Flyer

Free Software in the Curriculum

Some Suggestions and Evaluation Criteria

Background: Like many educational organizations, Scholastech acquired public domain resources because we couldn't afford ten copies of expensive, licensed software. Furthermore, we needed to share our applications with colleagues and students without worrying about the legal implications of distribution. Although we regularly purchase and use licensed software for selected tasks, most of our organizational and instructional computing is based upon the public domain and user-supported software listed in this catalog. Not only are some of these packages good, they are very often superlative in performance and extensibility (i.e. the degree to which you can customize the package). In the following brief discussion, we would like to offer you some observations on the state of free software for education and some practical guidelines for evaluating these resources.

Basic Distinctions: The axiom "you get what you pay for" has inhibited many educators from exploring the benefits of free software. We offer a new axiom: "not everything wrapped in cellophane is gold." Commercially copyrighted and licensed software is as uneven in performance and support as any public domain program; paying $295 is no guarantee that your specific questions or needs will be addressed by the distributor. Free software is available under many categories: public domain, shareware, testware, and siteware. Distinguishing what each category implies is essential for your evaluation of free software.

Public Domain software: Technically, software in the public domain is not copyrighted, cited by the author(s), or supported by anyone in particular. There may be an author's name listed in the source code (if available) or on an opening screen, but present users have no way to contact the author. Such programs may be freely copied, modified, and distributed without restriction. Some commercial programs in fact were built upon public domain source code.

Freeware: Some programs have been copyrighted or publicly claimed by the author to protect and monitor the integrity of the program. From year to year users who modify the source code are urged to archive their changes with the author. Major development projects such as Richard Stallman's *GNU EMACS* or our own *Scholastech Authoring System* may be freely copied with the understanding that the authors wish to monitor the quality and integrity of subsequent source code revisions. Freeware is usually well-documented, providing some guidelines for distribution and modification. Check the source code or accompanying documentation for each program.

Shareware: Some of the most creative programmers have developed a form of licensing called "shareware." The program is copyrighted by the author, licensed to the user under certain conditions, and (for registered users) supported by the author. Although the terms of each shareware license differ slightly, generally the user is free to copy the program for the purposes of evaluation. If the program proves to be useful (a tricky definition), the user is encouraged to register his or her copy with the author for a nominal fee. In exchange for registration, the user generally receives a hardcopy manual, free updates, and phone support. The success of such programs as *PC-Write*, *PC-Outline*, and *PC-File* have demonstrated the viability of the shareware concept. Read the documentation carefully. If an author says that the program may be freely copied for purposes of evaluation, that constitutes a license to copy and distribute that program for evaluation. Do not feel hesitant about copying shareware; the authors make their money because you and I copy their programs for colleagues and students who (in theory) send in their fees.

Testware software: As a derivation of shareware licensing, testware restricts your ability to make copies of the program. For example, you can make a backup copy of *Analyticalc* for your own evaluation, but you are not to copy it freely for others. If you find a testware program inadequate for your needs, you are supposed to erase it. Too many subtle restrictions about the copying and sharing of free software prove counterproductive. If an author puts bizarre restrictions on the program and you feel hesitant to call it true shareware, let it go. We at Scholastech have not found any testware software clearly superior in performance to genuine shareware programs.

Siteware software: As another shareware variation, siteware provides for unrestricted copying between individuals, but restricted distribution within organizations. The excellent communications program *ProComm*, for example, encourages free copying by individual users, but requires that a company or university seek a site license before distributing the program to more than a few evaluators. Traditional shareware authors offer site licenses for the sake of support; siteware authors offer an individual license to copy in order to land site licenses. If you are an administrator, the distinction between shareware and siteware is important. If you are an individual user, just be certain to find the section of documentation delineating your right to make copies.

When you receive a free program from a local user group or distribution group like Scholastech, you can be confident (within reason) that the legal status of the program has been checked out prior to distribution. If you download a program from a bulletin board or network, you will have to do some verification on your own. In any event, *do not be reluctant to copy any program that says you can make*

free copies. There is often a healthy tension at colleges between central computing centers and the individual user on this issue of copying; in all cases consider the license provided with the software to be literal.

Current Models For Selecting Curriculum Software: As the personal computer emerged as the mainstay for curriculum computing in higher education, administrators and professors developed three loosely–contiguous models for selecting PC software. The ***productivity model*** reflects a concern for campus–wide software coherence. Instructors and students in one department should be performing word processing and spreadsheet work with software compatible with other departments. Although this model is particularly successful at schools new to curriculum computing, longrange policies based upon "standard software" prove stifling after the first year or two. In the overall complexion of campus computing, there should be complementarity––campus–wide productivity packages leading students toward more discipline–specific resources. The ***professional model*** reflects a concern for student placement in business. If new graduates will work in offices using XYZ commercial software, they should use XYZ software in the curriculum. The boundries between higher education and training are fuzzy here. Some educators make a legitimate case that exposure to software of professional performance and complexity is essential for the learning process. Others argue that exposing students to industrial–strength programs diverts time and energy from underlying concepts. The ***development model*** stresses the problem–solving aspects of curriculum computing. If students are being exposed to computers to solve problems, they should be able to select the appropriate software addressing appropriate aspects of the problem. Instructors therefore need a library of flexible software analogous to their library of books. Instructors need to be able to customize the capabilities of a program to reflect discipline–specific methodology and data.

Where Does Free Software Fit? Colleges rarely select and support software on the basis of one model. Most often the underclass curriculum is based upon ***productivity***, the upperclass upon ***professional***, and isolated pockets in the curriculum thrive on ***development***. Wherever you find yourself on this spectrum, there is free software relevant to your needs. Public domain and shareware software represents a "no–risk" environment for testing new ideas and evaluating new methods. If you are not a programmer, public domain software offers a greater degree of "authoring" potential than closed–architecture commercial programs. Most of all, software that can be copied freely encourages the free dissemination of ideas based upon that software. Educators cannot restrict the medium of exchange in curriculum computing (i.e. software) without restricting the volume and quality of exchange. Curriculum computing should ***enhance*** the process of exchanging information and solving problems, not ***restrict*** it to a pay–as–you–go policy. Toward the development and support of computing educators familiar with the benefits of free software Scholastech dedicates its energies and resources.

Educational Users –– Assessing Needs: Scholastech tests and supports several types of spreadsheets, text processors, and data base managers in recognition of the diverse needs of computing educators. The instructor considering a free software package to support a two–week module in some discipline simply cannot introduce a full–featured, sophisticated piece of software; the learning curve would be prohibitive. An little public domain spreadsheet like *Minicalc* may be appropriate for a limited exercise. Conversely, the instructor selecting a package to be used throughout a semester should consider time investment––students should not exhaust the potential of such a package in the first few hours. Although categories of academic users can be oversimplistic, some general groupings by interest and expertise will help you evaluate potential audiences for free software.

Basic Users: If you have no desire to learn a programming language or develop instructional macros, consider your role as a *basic user*. Free software that you can integrate with course material should be easy to use, relatively self–evident in operation, and (above all) well–documented. Programs such as *PC–Outline* or *PC–Calc* offer your students tremendous power without the entanglements of user–defined configurations and sophisticated macros. If you are comfortable using a friendly application program, your students will sense that and feel comfortable as well. Computer literacy and computer–based problem solving begin with the effective management of such basic programs.

Developers/Hackers: If you are comfortable with the operation of basic packages and yearn to develop new applications for your courses, free software can offer you considerable "authoring" potential. A word processor like *PC–Write*, for example, not only does a fine job on term papers, it also provides the instructor with "hooks" into its operation. By developing new "ruler files" or macros for *PC–Write*, an instructor can recreate many of the functions of *Writer's Workbench*, design analytic functions for scanning poetry, or construct interactive lessons on grammar. You do not have to know Pascal or "C" in order to develop your own sophisticated applications. Any free software package with macro capabilities will support such original development. Programs with windowing capabilities or a DOS shell will support the development of even more sophisticated applications.

Programmers: Even with macros and windows, some application programs will not support very discipline–specific development. This is especially true in disciplines requiring extensive graphic representation. If you know a programming language like Pascal, BASIC, or "C," many free software packages provide source code. You can, in effect, modify the source code to reflect very specialized needs. If you are interested in learning a computer language that performs high–level tasks, consider such public domain languages as *ICON*, *PD–Prolog*, or *XLISP*. In these languages you can develop some pretty powerful

e. multiple–files and windows?
f. DOS Shell?

3. User Interface
   a. logical and consistent on–screen formatting?
   b. menus/options displayed?
   c. on–line help?
   d. function key support?

4. Development Potential
   a. macro capabilities?
   b. clear development guidelines?
   c. interactive capabilities?
   d. source code provided?

Evaluations Welcome: Many patrons and supporters of Scholastech request that we publish a detailed comparative evaluation of each software package in the library. Although this is a tempting prospect, we believe that this would be counterproductive in two respects. First, recommendations tend to gather some subtle sanctity. If most Scholastech members would recommend *PC–Write* over *EMACS* for composition classes, this would discourage some educators from examining *EMACS* at all. It's too easy to confuse recommendations with standards. Yes, a word processor for the curriculum should have certain standard features, but no one package should be considered a standard. Secondly, Scholastech develops and maintains a library of free software so that educators can order and evaluate several packages at practically no cost. The process of evaluating free software is in itself educational and collegial. If an instructor must articulate why package B is less appropriate than package A, he or she must get involved in an evaluation of the teaching context itself. He or she has opportunity to discuss the relative merits of software with colleagues and administrators. The adoption of curriculum software without such general discussion and evaluation is criminal.

In subsequent issues of this software catalogue we would like to publish some of the comparative evaluations which educators have sent us. If you have copies of this free software or other packages not catalogued here, please send us a brief (1–2 pp.) evaluation of their relative merits. We will try to publish every evaluation which we receive.

Conclusion: Considered from any number of perspectives, free software makes sense for individual instructors and for academic departments. In your efforts to develop discipline–specific applications, please consider Scholastech a material and professional resource.

Cynthia D. Leeming

QUICK REFERENCE LISTING
SCHOLASTECH PUBLIC DOMAIN RESOURCES -- MS-DOS

ABBREVIATIONS

DISK .... the disk designation for ordering
S/ource.. source code included--C, B/asic, P/ascal
VERS..... version of the package
COM ..... Compatibility Index..... (generally speaking)

1 - PC-DOS Clones	- IBM PC/AT	- Kaypro 16
	- ATT PC6300/Plus	- Panasonic SP
	- Data General One	- Visual Comm.
2 - MS-DOS Comp.	- TI Professional	- Epson QX-PC
3 - MS-DOS PCs	- DEC Rainbow 100	- Fujitsu Micro16s
	- Wang PC	

THUS: a "3" runs on any PC, a "2" on most, a "1" on a clone

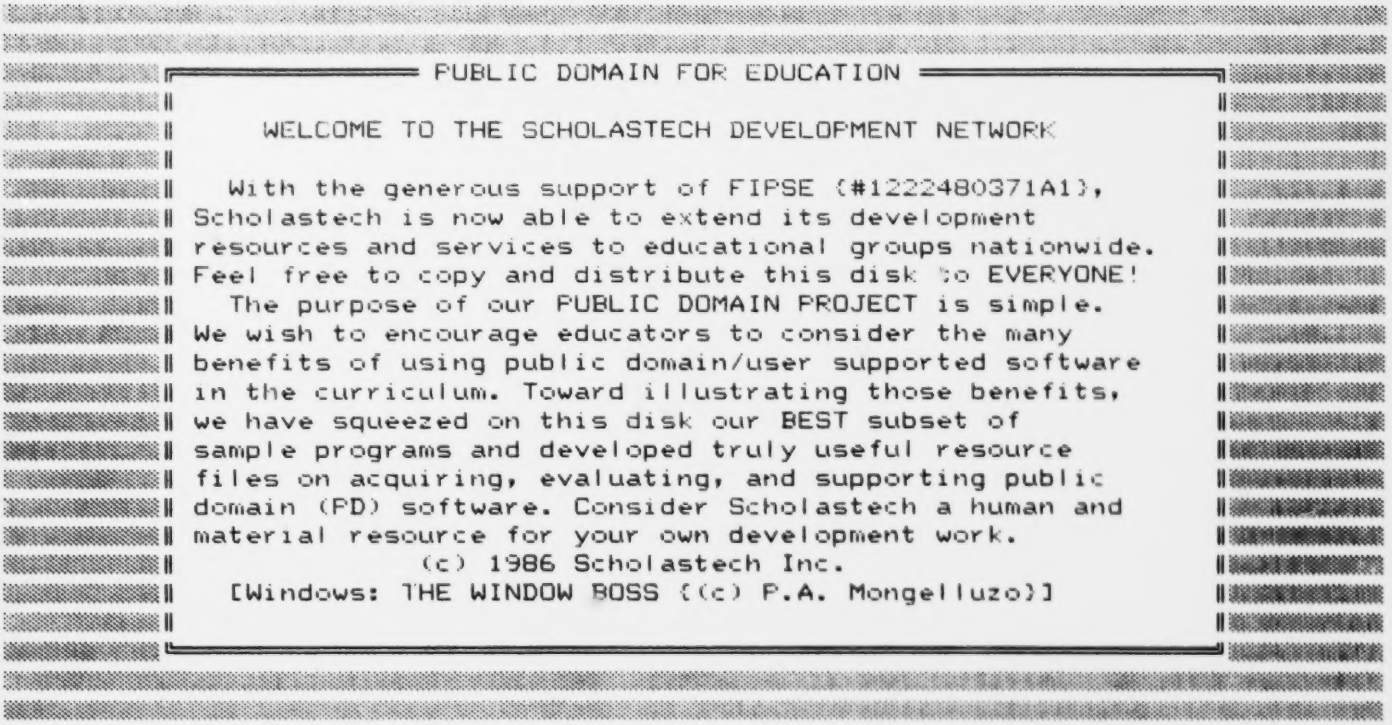
DOC ..... quality of documentation included on disk (A-F)
ST ..... legal status--US (user-supported), PD (public domain)

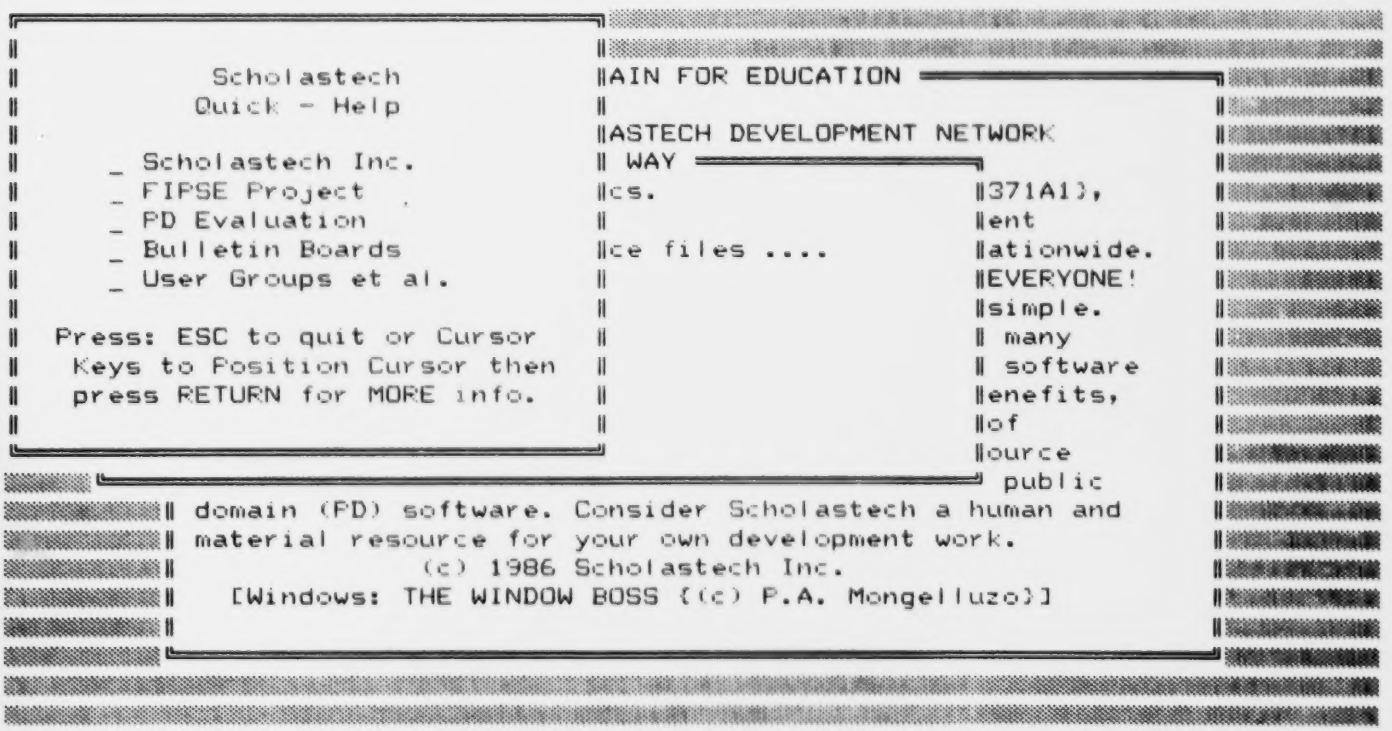
DISK	PROGRAM	VERS	S	COM	DOC	ST	DESCRIPTION
AI1	XLISP	1.4a	C	3	B	US	AI LANGUAGE INTERPRETER
AI2	PPEI-AI		B	2	B	US	INFERENCE SHELL (NEEDS DB3/4)
AI3	PROLOG	1.32		3	B	US	A.D.A. PROLOG INTERPRETER
AI4	ESIE	1.0		2	C	US	QUESTION/ANSWER INFERENCE SHELL
BUS1	DEPREC		B	2	D	US	DEPRECIATION CALCULATOR
	IPM	1.7		2	B	US	PROJECT SCHEDULER -- OK
	TASKPLAN	1.0	B	2	D	US	SIMPLE PROJECT SCHEDULER
BUS2	LIGHTYR	1.0		1	B	US	FORECASTING SIMULATIONS -- OK
BUS3	PC-GL	1.3		1	B	US	GENERAL LEDGER -- REASONABLE
COM1	MODEM7			2	C	PD	GENERAL COMMUNICATIONS
	PC-TALK		B	2	B	US	VERSATILE COMMUNICATIONS
	TERM		C	3	C	PD	TERMINAL EMULATOR - LATTICE
COM2	KERMIT	2.28		1-3	C	PD	TERM / TRANSFER - VERSATILE!!
COM3	COM-SHEL	1.6		3	A	US	UNIX-LIKE COMMUNICATIONS SHELL
COM4	PCVT100	8.3		1	A	US	VT100 EMULATOR/TRANSFER
COM5	QMODEM	2.0		2	A	US	VERSATILE & FAST TERM/TRANSFER
DB1	ABCFILE			2	B	US	FLEXIBLE DATA BASE MANAGER
DB2	PC-FILE	4.0		3	A*	US	POWERFUL, FLEXIBLE DBMS
DB3	PPEI-DB1		B	3	A	US	RELATIONAL, PROGRAMMABLE DBMS
DB4	PPEI-DB2		B	3	A	US	PPEI SUPPORT FILES
DB5	MAIL		B	3	C	US	MAILING LISTS & LABELS
	SDB	2.0	C	3	A*	PD	COMMAND-ORIENTED DBMS
DB6	BOOKINDX			2	D	PD	BIBLIOGRAPHIC RETRIEVAL
	MAILIST1		B	3	C	US	MAILING LISTS -- LABELS
	PC-LIB		B	3	C	US	DISK DIRECTORY MANAGEMENT
DB7	JBY5	3.0		2	B	US	SIMPLE RECORDS MANAGER
DB8	PBASE			1	B	US	RELATIONAL DATA BASE MGR.
DEV1	BAT-EXTN	2.0		2	A	US	EXTENDED BATCH LANGUAGE
	NEWKEY	1.1		1	A	US	PROKEY CLONE -- DEFINE KEYS
	SMALL-C	1.1	C	3	C	PD	PUBLIC DOMAIN C COMPILER
DEV2	ASDKEYS	1.0		2	A	US	KEY DEFINITIONS -- HOT!
	ICON	5.9		3	C	US	LIKE SNOBOL4, TEXT LANGUAGE

DISK	PROGRAM	VERS	S	COM	DOC	ST	DESCRIPTION
DEV3	CSHELL		C	3	B	US	UNIX-LIKE SHELL FOR MS-DOS
DEV4	F-CONSOL	1.09		1	B	US	REPLACE KEYBD/SCREEN SYSTEM
GR1	CHEAPCAD	1.1		1	C	US	FAST & DIRTY CAD; SOME BUGS
	DESIGNER	1.0	B	1	C	US	GRAPHICS AUTHORING SYSTEM
	PDRAW		B	1	D	PD	GRAPHICS AUTHORING SYSTEM
GR2	LADYBUG	1.0		1	B	US	LOGO-LIKE GRAPHICS AUTHORING
	TABLET		B	1	D	PD	GRAPHICS AUTHORING SYSTEM
GR3	PC-GRAPH	1.0		1	B	US	GRAPHICS FOR PC-FILE & PC-CALC
GR4	PC-KEY1	2.0		1	B	US	BASIC, CAPABLE CAD EDITOR
GR5	PC-KEY2			1	B	US	SUPPORT FILES FOR PC-KEY1
GR6	PRESENT	5.1		1	B	US	SLIDE-PRESENT FOR PC GRAPHS
QA1	EPISTAT	3.0	B	2	B	US	FULL-FEATURED STATS/GRAPHICS
	STATPAK		B	2	C	US	BASIC STATISTICS ROUTINES
QA2	PC-PAD	1.3	B	1	B	US	TEXT & SPREADSHEET PROCESSOR
QA3	FREECALC	1.0		2	B	US	FULL-FEATURED SPREADSHEET
QA4	ECSTAT		B	2	B	US	ECONOMETRIC ANALYSIS SYSTEM
	KLP	1.9		1	D	US	KINETIC LINEAR PROG. SYSTEM
QA5	CALC		C	3	B	PD	12-FUNCTION CALCULATOR
	CALENDAR			3	NA	PD	MONTHLY DISPLAY
	MINICALC			2	B	PD	ONE-SCREEN SPREADSHEET
QA6	PC-CALC	2.0		2	B	US	BUTTONWARE SPREADSHEET - GOOD!
QA7	SOLVER!			2	C	US	TK-SOLVER WORK-ALIKE
TXT01	EMACS	3.62	C	3	A*	PD	CLASSIC!! WINDOWS & BUFFERS
	NRO		C	3	A	PD	TEXT FORMATTER/PRINTER - UNIX
TXT02	PC-WRITE	2.6		1	A*	US	FULL-FEATURED WORD PROCESSOR
TXT03	C-EDITOR		C	2	C	PD	VERSATILE SCREEN EDITOR
	ROFF		C	3	B	PD	TEXT FORMATTER
TXT04	EDIT V1			1	B	US	CAPABLE EDITOR: WWRAP, MACROS
TXT05	DVED	6.02		2	B	US	FULL-SCREEN EDITOR -- NICE
	FMT		P	3	B	PD	TEXT FORMATTER / PRINTER
	LIST552	5.52		1	A	US	READ-ONLY EDITOR / PRINTER
TXT06	FRED			1	B	US	POPULAR ALTERNATIVE TO EDLIN
	WPK			1	B	US	WORD PROCESSOR FOR KIDS: BASIC
TXT07	SEDTV12			RBW	A	PD	RAINBOW WORD PROCESSOR - NICE!
TXT08	PC-FOIL			1	B	US	INTEGRATED TEXT/GRAPHICS EDITR
TXT09	PC-OUTLN	1.08		1	A	US	IDEA/OUTLINE PROCESSOR - GREAT
TXT10	PC-JOVE		C	1	B	PD	ENHANCED EMACS--WRITERS' LOVE
TXT11	MAX			1	B	US	FAST, EMACS-LIKE, SMALL SIZE
	PROINDEX	1.2		3	B	US	INDEX & CROSSREF. BY KEYWORDS
	SPELL2			3	B	US	BASIC SPELLING CHECKER 20K DIC
	VINCENT		C	1	B	US	ANSI TEXT/GRAPHICS EDITOR
UTL1	VF150			1	A	US	POWERFUL DISK/FILE MANAGER
	WINDOW4C			3	A	US	MULTI-WINDOW LIBRARY / LATTICE
UTL2	DESKMATE			1	A	US	SIDEKICK WORK-ALIKE
	PCWINDOW			1	B	US	SIDEKICK WORK-ALIKE

TO ORDER DISKS: send $5/disk to SCHOLASTECH INC.
P.O. Box 1768
Cambridge MA 02238

SAMPLE RESOURCE SCREENS
for IBM/ATT PC-compatibles


PUBLIC DOMAIN FOR EDUCATION
WELCOME TO THE SCHOLASTECH DEVELOPMENT NETWORK
With the generous support of FIPSE (#1222480371A1),
Scholastech is now able to extend its development
resources and services to educational groups nationwide.
Feel free to copy and distribute this disk to EVERYONE!
The purpose of our PUBLIC DOMAIN PROJECT is simple.
We wish to encourage educators to consider the many
benefits of using public domain/user supported software
in the curriculum. Toward illustrating those benefits,
we have squeezed on this disk our BEST subset of
sample programs and developed truly useful resource
files on acquiring, evaluating, and supporting public
domain (PD) software. Consider Scholastech a human and
material resource for your own development work.
(c) 1986 Scholastech Inc.
[Windows: THE WINDOW BOSS (c) P.A. Mongelluzo]



SCHOLASTECH PUBLIC DOMAIN
R & D DISK v. 1.3 (FIPSE)
A>welcome (to start)

PUBLIC DOMAIN SOFTWARE

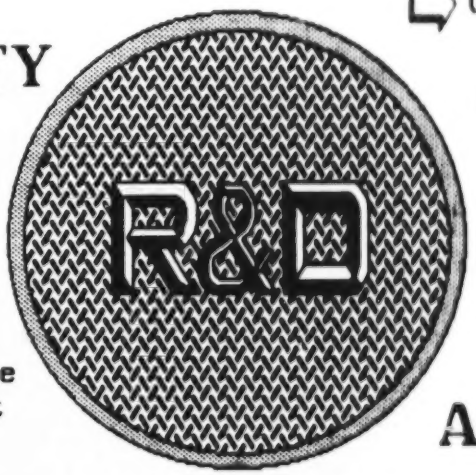
IT'S ALL YOURS !

R&D is an extraordinary collection of public domain software for faculty, students, and administrators in colleges and universities around the nation.

Developed through funding from FIPSE (Funds for the Improvement of Post Secondary Education), R&D public domain software has been selected to aid in text manipulation and file handling.

OVER TWENTY PROGRAMS

Over twenty programs in the collection cover a host of academic needs, from printing through creation of a RAM disk. Here are some of the programs you get with R&D:



- 3BY5: Manages text & data -- full featured
- Crypt: Encrypts top secret files
- EMACS: Processes text -- full featured
- List 565: Browses files
- LQ-Print: Prints Epson Letter quality
- Move: Moves files from directory to directory
- Kermit: Allows micros to "talk" to mainframes
- PD Sideways: Prints spreadsheets sideways
- QuickSort: Sorts data -- very sophisticated
- RamDisk: Creates virtual disk in memory
- RenDir: Renames a directory
- Snapshot: Stores screen images
- UnDel: Undeletes erased floppy disk files
- UnWordStar: Makes ASCII files from WordStar files
- VF150: Manages Directories

COPY AND SHARE

This introductory disk has been distributed directly to over a thousand educators. If each person copies it for several colleagues, many thousands of educators in the social sciences, humanities, and quantitative sciences will benefit.

For a copy of R&D, please send $5 (diskette, handling, & mailing cost) to

Scholastech, Inc.
Box 1545
Cambridge, MA 02238

R&D runs on all IBM and Compatible computers.

SCHOLASTECH OF CAMBRIDGE